Part1 Problems 1 to 7

SPPU DBMS PRACTICALS - SET II
Part 1: Problem Statements 1 - 7
==========

Problem Statement 1 — CRUD (MongoDB)

Collection: Student(Roll_No, Name, Class, Marks, Address, Enrolled_Courses)

Commands:
use College;
db.createCollection("Student");

db.Student.insertMany([
{ Roll_No: "A01", Name: "Vivek", Class: "SE", Marks: 65, Address: "Pune", Enrolled_Courses: ["DBMS","OS"] },
{ Roll_No: "A02", Name: "Priya", Class: "TE", Marks: 78, Address: "Mumbai", Enrolled_Courses: ["DBMS","TOC"] },
{ Roll_No: "A03", Name: "Rohan", Class: "BE", Marks: 45, Address: "Nashik", Enrolled_Courses: ["CN","SE"] },
{ Roll_No: "A04", Name: "Sneha", Class: "SE", Marks: 88, Address: "Pune", Enrolled_Courses:
["DBMS","TOC","SE"] },
{ Roll_No: "A05", Name: "Kiran", Class: "TE", Marks: 52, Address: "Nagpur", Enrolled_Courses: ["OS","CN"] },
{ Roll_No: "A06", Name: "Amit", Class: "BE", Marks: 19, Address: "Solapur", Enrolled_Courses: ["DBMS","AI"] },
{ Roll_No: "A07", Name: "Manisha",Class: "SE", Marks: 72, Address: "Kolhapur", Enrolled_Courses: ["TOC","AI"] },
{ Roll_No: "A08", Name: "Sarthak",Class: "TE", Marks: 94, Address: "Pune", Enrolled_Courses: ["DBMS","ML"] },
{ Roll_No: "A09", Name: "Neha", Class: "BE", Marks: 34, Address: "Latur", Enrolled_Courses: ["CN","DBMS"] },
{ Roll_No: "A10", Name: "Rutuja", Class: "SE", Marks: 49, Address: "Satara", Enrolled_Courses: ["OS","TOC"] }
]);

Queries:
1. db.Student.find({ Enrolled_Courses: { $in: ["DBMS","TOC"] } }, { Name:1, _id:0 });
2. db.Student.find({ $or: [{ Marks: { $gt: 50 } }, { Class: "TE" }] }, { Roll_No:1, Class:1, _id:0 });
3. db.Student.updateOne({ Roll_No: "A10" }, { $set: { Name:"Rutuja Patil", Class:"TE", Marks:81, Address:"Pune",
Enrolled_Courses:["DBMS","AI","CN"] } });
4. db.Student.find({}, { Name:1, Marks:1, _id:0 }).sort({ Marks:-1 }).skip(2).limit(2);
5. db.Student.deleteMany({ Marks: { $lt: 20 } });
6. db.Student.deleteOne({});

Problem Statement 2 — DDL using MySQL/Oracle

Tables:
Customer(CustID PK, Name, Cust_Address, Phone_no, Email_ID, Age)
Branch(BranchID PK, Branch_Name, Address)
Account(Account_no PK, BranchID FK, CustID FK, open_date, Account_type, Balance)

Create tables:
CREATE TABLE Customer (
CustID INT PRIMARY KEY,
Name VARCHAR(100),
Cust_Address VARCHAR(200),
Phone_no VARCHAR(20),
Email_ID VARCHAR(100),
Age INT
);

CREATE TABLE Branch (
BranchID INT PRIMARY KEY,
Branch_Name VARCHAR(100),
Address VARCHAR(200)
);

CREATE TABLE Account (
Account_no INT PRIMARY KEY,
BranchID INT,
CustID INT,

```
open_date DATE,
Account_type VARCHAR(20),
Balance DECIMAL(12,2),
FOREIGN KEY (BranchID) REFERENCES Branch(BranchID),
FOREIGN KEY (CustID) REFERENCES Customer(CustID)
);
```

Insert sample data:

```
INSERT INTO Customer VALUES
(101,'Vivek','Pune','9876543210','vivek@gmail.com',21),
(102,'Priya','Mumbai','9876501234','priya@gmail.com',22),
(103,'Rohan','Nashik','9876549876','rohan@gmail.com',23),
(104,'Sneha','Nagpur','9876512345','sneha@gmail.com',24),
(105,'Kiran','Kolhapur','9876598765','kiran@gmail.com',25);
```

```
INSERT INTO Branch VALUES
(1,'Main Branch','Pune'),
(2,'Central Branch','Mumbai'),
(3,'North Branch','Nashik');
```

```
INSERT INTO Account VALUES
(1001,1,101,'2018-08-16','Saving',25000),
(1002,2,102,'2018-02-16','Loan',50000),
(1003,3,103,'2018-08-16','Saving',30000),
(1004,1,104,'2018-02-16','Loan',45000),
(1005,2,105,'2019-05-18','Saving',22000);
```

Tasks & Queries:

```
3. CREATE VIEW Saving_Account AS SELECT c.*, a.Account_no, a.open_date, a.Account_type, a.Balance FROM
Customer c JOIN Account a ON c.CustID = a.CustID WHERE a.Account_type='Saving' AND a.open_date =
'2018-08-16';
4. UPDATE Customer SET Cust_Address = 'Pune' WHERE CustID = 103;
5. CREATE VIEW Loan_Account AS SELECT c.*, a.Account_no, a.open_date, a.Account_type, a.Balance FROM
Customer c JOIN Account a ON c.CustID = a.CustID WHERE a.Account_type='Loan' AND a.open_date =
'2018-02-16';
6. CREATE INDEX idx_customer_custid ON Customer(CustID);
7. CREATE INDEX idx_branch_branchid ON Branch(BranchID);
8. CREATE SEQUENCE cust_seq START WITH 106 INCREMENT BY 1 NOCACHE;
9. CREATE SYNONYM Cust_info FOR Branch;
```

Problem Statement 3 — Unnamed Block (PL/SQL)

Schema: Employee(emp_id, emp_name, DoJ, salary, commission, job_title)
Salary_Increment(emp_id, new_salary)

PL/SQL unnamed block:

```
DECLARE
v_emp_id NUMBER := &amp;emp;_id;
v_salary NUMBER;
v_doj DATE;
v_years NUMBER;
v_new_salary NUMBER;
BEGIN
SELECT salary, DoJ INTO v_salary, v_doj FROM Employee WHERE emp_id = v_emp_id;
```

```
v_years := FLOOR(MONTHS_BETWEEN(SYSDATE, v_doj)/12);
```

```
IF v_years > 10 THEN
v_new_salary := v_salary * 1.20;
ELSIF v_years > 5 THEN
v_new_salary := v_salary * 1.10;
ELSE
v_new_salary := v_salary * 1.05;
END IF;
```

```
UPDATE Employee SET salary = v_new_salary WHERE emp_id = v_emp_id;
INSERT INTO Salary_Increment(emp_id, new_salary) VALUES (v_emp_id, v_new_salary);
COMMIT;
DBMS_OUTPUT.PUT_LINE('Salary updated to ' || v_new_salary);
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Employee not found: ' || v_emp_id);
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
ROLLBACK;
END;
/
```